2

NPS-MA-93-016

NAVAL POSTGRADUATE SCHOOL

Monterey, California

NEURAL NETWORK IDENTIFICATION OF KEYSTREAM GENERATORS

by

Jeffery J. Leader
LT James E. Heyman

Technical Report For Period
January 1993 - March 1993

Prepared for: Naval Postgraduate School
Monterey, CA 93943

93

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943

This report was prepared in conjunction with research conducted for the Naval Postgraduate School and funded by the Naval Postgraduate School.

This report was prepared by:

[illegible]

Jeffery J. Leader
Assistant Professor of Mathematics

[illegible]

LT James E. Heyman

Reviewed by:

Richard Franke

RICHARD FRANKE
Chairman

Released by:

[illegible]

PAUL J. MARTO
Dean of Research

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704 0188

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATIO DOWNGRADING SCHEDULE	
4 PERFORMING ION REPORT NUMBER(S) NPS-MA-93-016	5 MONITORING ORGANIZATION REPORT NUMBER(S) NPS-MA-93-016
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable) MA	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City nd ZIP Code) Monterey, CA 93943	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943
8a NAME OF FUNDING/SPONSORING ORGANIZATION Naval Postgraduate School; 8b OFFICE SYMBOL (If applicable) MA	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER OM,N
8c ADDRESS (City d ZIP Code) Monterey, CA 93943	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Neural Network Identification of Keystream Generators

12 PERSONAL AUTHOR(S) Jeffery J. Leader, James E. Heyman

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month Day)	15 PAGE COUNT
Technical	FROM 1-93 TO 3-93	6 May 93	15

16 SUPPLEMENTARY NOTATION

17			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	G	JB GROUP	Keystream generators

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Applications such as stream ciphers and spread spectra require the generation of binary keystreams to implement, and the simulation of such keystreams to break. Most cryptanalytic attacks are of the known generator type, that is, they assume knowledge of the method used to generate the keystream. We show that a neural network can be used to identify the generator, and in some cases to simulate the keystream.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT [✓ UNCLASSIFIED/UNLIMITED] ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Jeffery J. Leader	22b TELEPHONE (Include Area Code) 408-656-2335; 22c OFFICE SYMBOL MA/Le

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

Neural Network Identification of Keystream Generators

LT James E. Heyman and Jeffery J. Leader

INTRODUCTION

There are many applications that require the generation of pseudorandom bitstreams, that is, sequences of zeros and ones that meet certain criteria. A partial list of these applications includes coding, communications, spread spectrum techniques (both for frequency allocation and security), simulations, and security access.

In addition to the typical measures of pseudorandomness, applications dealing with communications security impose the additional requirement that any attempt to retrieve the method of generation should be extremely difficult. For these cases it is useful to move beyond the scorecard of specific tests to a "larger" definition of a suitable sequence as one in which no amount knowledge of previous bits gives an indication as to what the next bit will be.

At least philosophically, it should be clear that any repeatable scheme that we as humans can come up with has to be deterministic (i.e. nonrandom) and therefore, at some level, predictable. It could be that the period is very large but that is not the point. The point is that it is logically possible.

An immediate reaction to the fact that pseudorandom sequences are being generated for security related applications is that it seems like a good idea to develop methods to attack them.

Eventually the goal is to be able to accurately predict future bits based on previous ones (assuming that through known plaintext or some other method we are able to retrieve pure keystream). Our preliminary goal, however, is a bit more modest: To determine the nature of the generator that was used to generate a particular bitstream. Although modest this aim is far from trivial for while there is much documentation concerning how to attack a sequence if the type of generator is known there is no obvious way of deciding which generator was used except by trial and error. As such, due to the prevalence of shift registers, the standard strategy is to attempt to simulate the sequence using Berlekamp-Massey (assuming a reasonable linear complexity) and if that doesn't work then to try something else. As more generation schemes appear this method becomes rather tedious and inefficient. For our purposes we restricted our investigation to three major generation methods: Shift registers, a quadratic planar map, and the linear congruential generator.

METHODOLOGY

Our proposed method of attack involves the use of a neural net. Specifically, we used the BrainMaker neural net software package to set up a back propagation model with fixed training and testing parameters which was used to test a series of n bits to predict one bit with n running from one to 25. It must be noted that throughout the experiments all of parameters such as learning rate, momentum, test rate, etc. were fixed. This was done to demonstrate the viability of the scheme in general (as opposed to finding specific solutions to specific coding problems). For all schemes bitstreams of length 1000 were used.

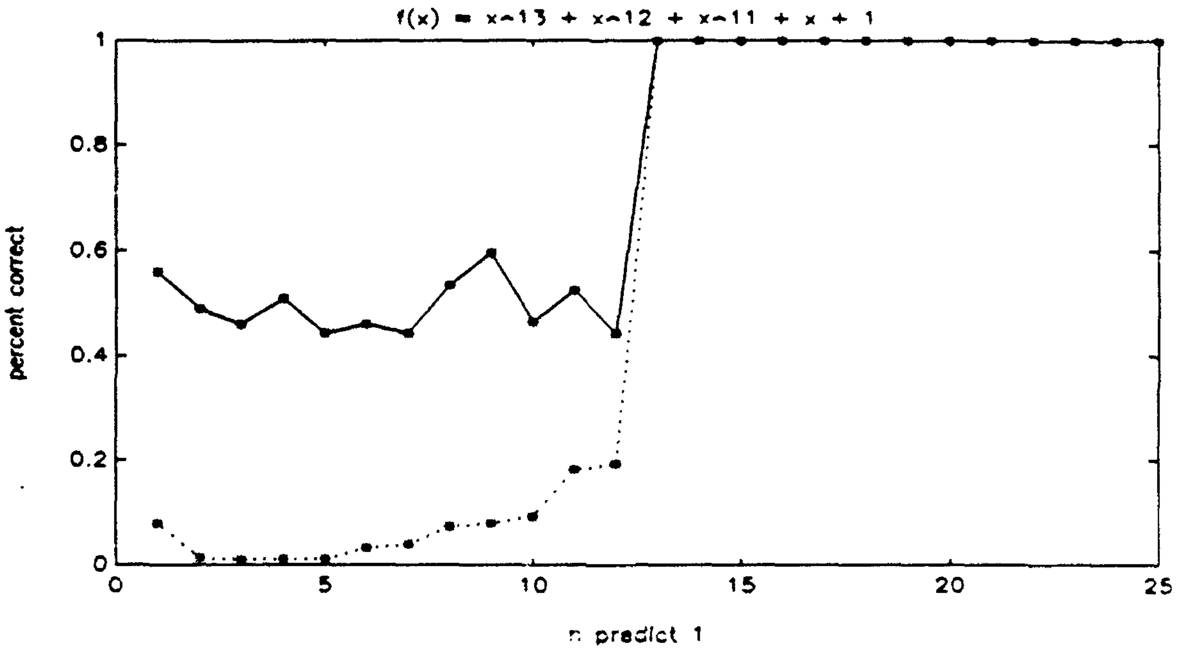
The basic idea is that for each family of generation schemes this process results in a characteristic training and testing curve that serves as a "fingerprint" of the process. The implication is that given a bitstream from an unknown source this attack will result in the determination of the family of generation schemes from which it came.

SHIFT REGISTERS

The most common scheme for generating pseudorandom bitstreams is the linear feedback shift register (LFSR), the details of which can be found in Golomb (1982). For our immediate purposes all that needs to be known is that an appropriately wired n-stage shift register will generate a binary sequence of length 2^n-1. This results in the favorable result that a fairly small shift register with, for example, only 64 stages can produce a sequence that is approximately 1.8×10^{19} bits long. Intuitively it would seem that such a sequence could baffle even the most powerful supercomputers but it turns out that, by utilizing the Berlekamp-Massey or Zeigler algorithm, a LFSR can be fully simulated if only 2n sequential bits are known. (This is not a difficult program and can be completed in 30 lines of MATLAB code.) Admittedly getting 2n bits of pure keystream might pose a problem but it surely is much less a problem than dealing with the entire sequence.

In as much as an effective algorithm exists for attacking shift register generated bitstreams it is reasonable to expect that any proposed new attack at least be able to match the efficacy of the known algorithms.

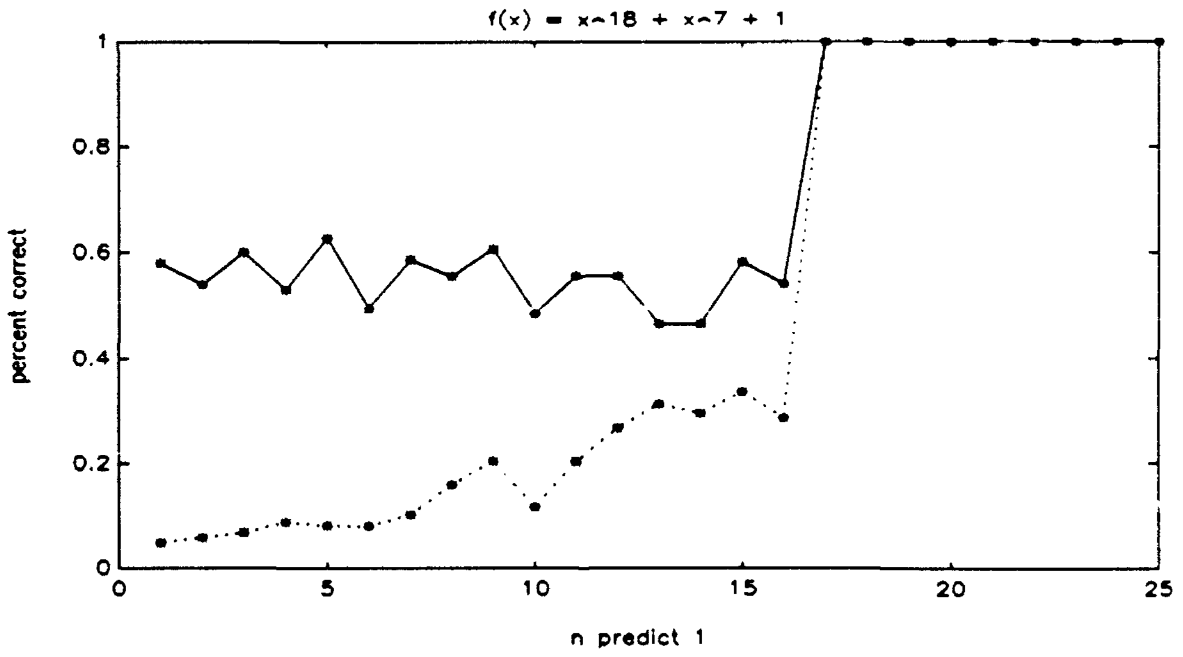
The first example is a LFSR defined by the function $f(x) = x^{13} + x^{12} + x^{11} + x + 1$. This function is primitive over GF(2) and as such the resultant sequence will have a full period of 8191. The following graph shows the result of applying the neural net in the manner described above.



As can be seen, the profile is similar to that of the linear complexity in the sense that the pattern is quickly determined (at n=13) and at that point any extra input bits are ignored. Of particular interest is how both the training (solid) and testing (dashed) lines remain relatively flat until the length of the shift register is reached, at which point both spike up to perfect training and testing. It is also noteworthy that once past that point that it usually took around 50 data passes to complete the training. Another facet is that it is possible to go back into the net and isolate which input bits have a direct effect on the output. The procedure is as follows: Label the bits as they correspond to the stages of an n-long shift register. Then, one by one, change the bits from zero to one (or one to zero as the case may be) and observe whether a given bit flips the output. If it does then the corresponding shift register stage of that bit has a direct effect on the output. Upon checking all of the input bits this results in the ability to recreate the polynomial associated with the actual LFSR that was used.

We can thus conclude that even though we used more bits than the Berlekamp-Massey algorithm would require we were able to generate the same results. More work needs to be done on the effectiveness of this method as the total number of bits analyzed is reduced in relation to the total length of the bitstream.

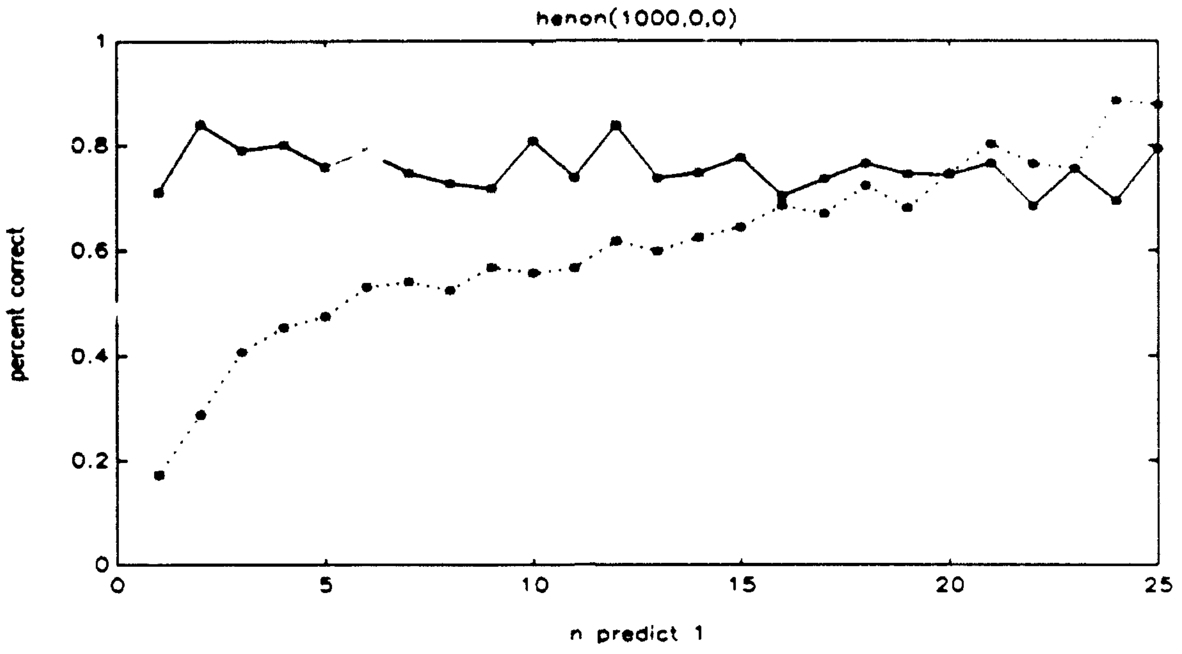
This experiment was then repeated on a LFSR based on the function f(x) = x^18 + x^7 + 1 with the resulting curves displayed below.

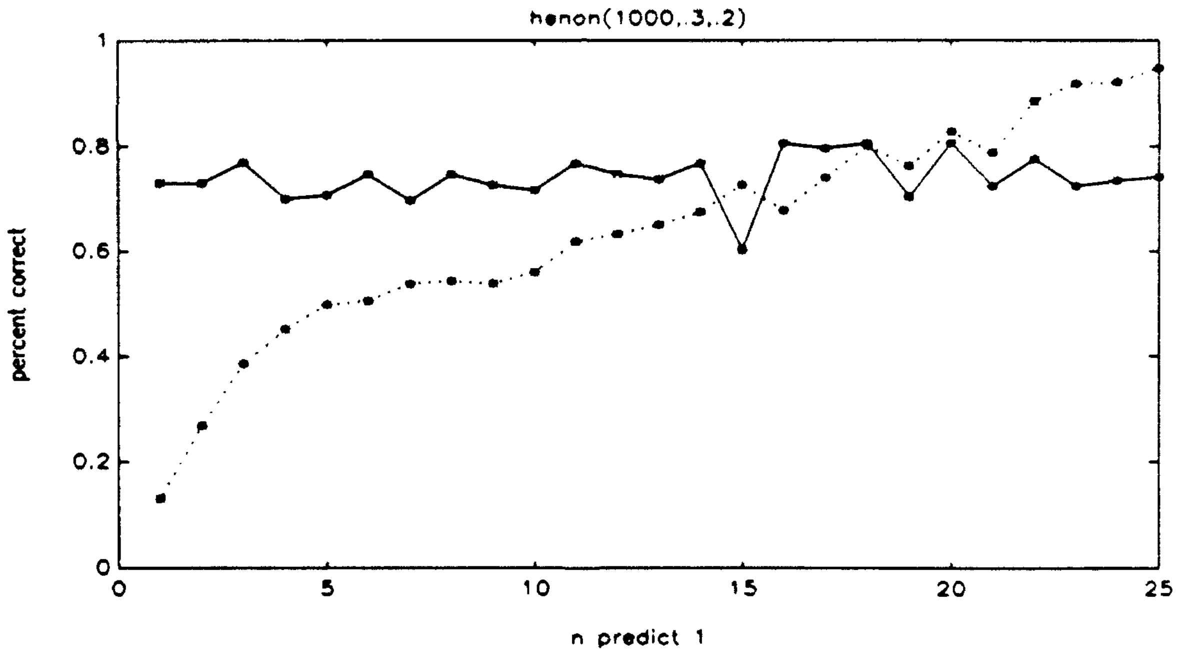


Note that the exact same fingerprint shows up. In conclusion, we believe that this combination of a training curve that stays low and flat combined with a testing curve that hovers around .5 until both jump up to 100% is indicative of the use of a linear shift register. In addition, by examining the net at the point of the jump the actual function can be retrieved and thus the shift register itself can be recovered.

QUADRATIC MAP

In previous work, Heyman (1993), the possibility of using a non-linear chaotic system to generate pseudorandom bitstreams was investigated. The process basically consists of following an orbit on the classic Hénon (1976) attractor and assigning a zero or a one based on whether the point was on the left or right side of a previously defined median point. Although there were some anomalies concerning the "runs" property the overall conclusion was that it did generate a reasonably pseudorandom sequence based on the larger definition of predictability. Although bitstreams of the same length as earlier (1000) were used these represent a minuscule percentage of the total bitstream length since this generation scheme has an approximate period of at least 10^{15} when generated on a personal computer. The following two graphs summarize the results of the neural net attack on bitstreams generated using this method.



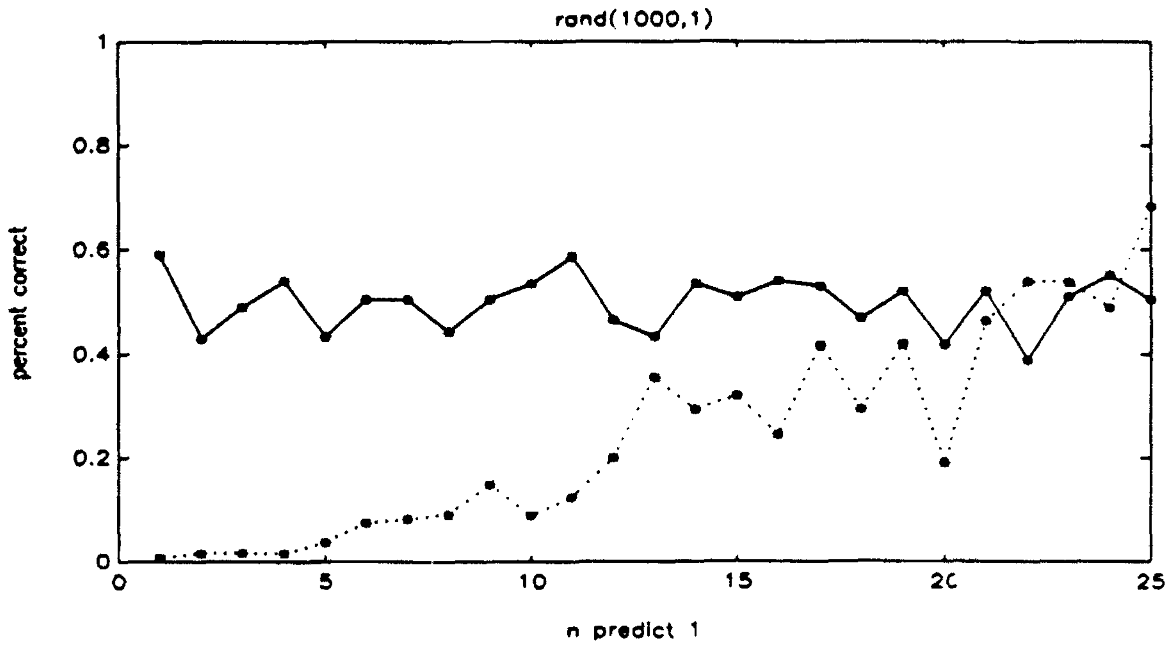


Notice that both show a constant testing level of about 75% while the training curve merges with the testing curve at n=15 and then breaks out above at n=25. This same behavior was observed on other Hénon bitstreams.

Inasmuch as the Hénon attractor is topologically conjugate to a wide class of quadratic mappings of the real plane (all those with constant Jacobian), it is expected that future work which applies this method of attack to other quadratic mappings will generate similar results. As such, at this point we are confident that this fingerprint is indicative of a bitstream which was generated with the Hénon scheme and we believe that a similar trace will be generated by any other planar quadratic mapping (used to generate a binary sequence as indicated in Heyman (1993)).

LINEAR CONGRUENTIAL

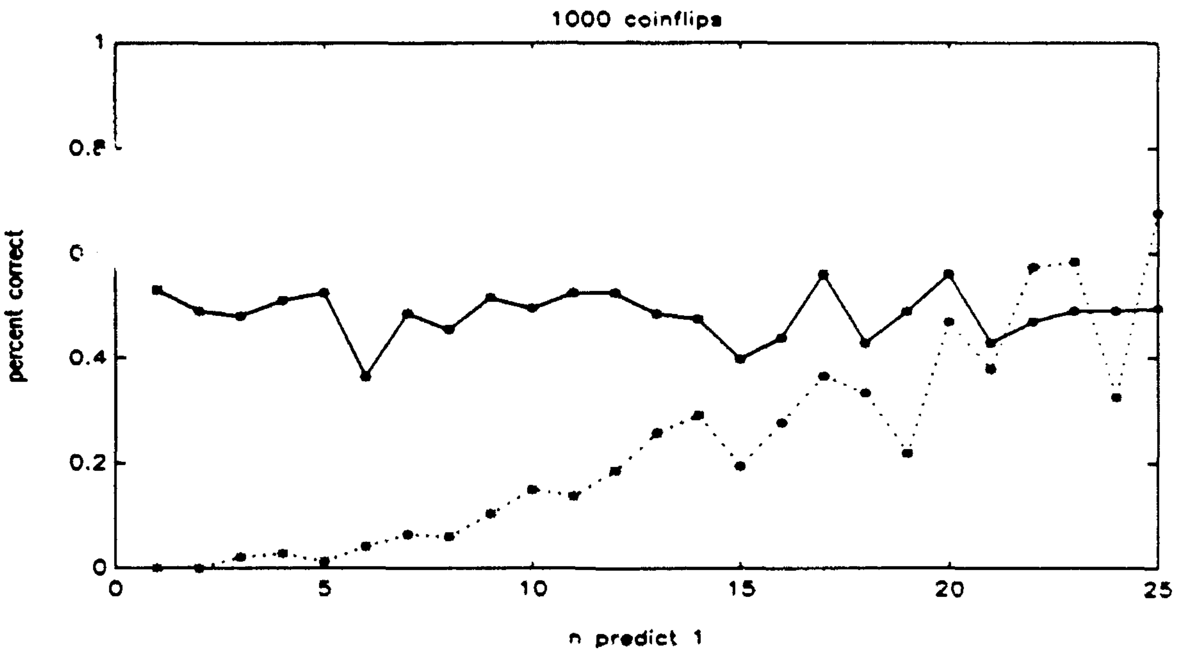
The next generation method used was based on the MATLAB random number generator, with the standard conversion from (0,1) to {0,1}. What makes the results surprising is that MATLAB uses a basic linear congruential generator which generally is not considered very sophisticated (although it is, of course, nonlinear; see Gillespie (1992)). Be that as it may, and whatever its other weaknesses are, as can be seen on the following graph, it does very well on this test (in terms of unpredictability). Nonetheless, it does have a distinctive fingerprint and thus the generation method can still be identified.



The distinguishing features of this fingerprint are a testing curve that hovers near 50% and a training curve that grows slowly until the two curves intersect in the range of n equal to 20 to 25.

It is quite arguable, philosophically or physically, that there are no macroscopic random processes. However, for most purposes, it is acceptable to consider the flip of a coin as a reasonably random event. For completeness, the neural net attack was then attempted on a bitstream generated by 1000 coin flips and the results follow.

As expected, the training lines hums right along around 50%. This is in keeping with the fact that even a neural net can't make progress on a truly random process. However, what is noteworthy is that the training line does seem to show a continual improvement. Since "progress" is purely a testing term this is not terribly



important except that it gives a secondary feature of the fingerprint, which is similar to the linear congruential generator discussed earlier.

CONCLUSION

As bit-stream generators become more sophisticated and schemes involving non-linear and chaotic processes become commonplace the traditional linear methods of attack will prove to be inadequate. Our proposal of using a neural net addresses this by utilizing it's inherent nonlinear workings to attack this nonlinear problem: We are fighting fire with fire. The graphs clearly indicate the value of this method for finding the generator. We mention that many standard attacks are of the "known generator" variety, and in this sense the neural net fills in a crucial gap between theory and practice by determining that generator. We have intentionally used the net in an unsophisticated manner, without varying the net parameters, in order to demonstrate the viability of this approach in general. The development of a library of fingerprints of known generators and of adapted artificial neural networks to identify them would certainly appear to be a worthwhile undertaking for the cryptanalyst.

Beyond the idea of generator characterization considered in this paper, and as evidenced by the testing results from the Hénon scheme, we further believe that this method will be effective in the actual prediction of bitstreams given some number of bits known to be correct. To achieve this, future work will have to include adaptive setting of the neural net software as well as the investigation of different types of neural nets. However, at the very least, we are absolutely convinced that neural nets can play a significant, and perhaps dominant, role in the process of attacking procedures that depend on pseudorandom bitstreams for their security.

BIBLIOGRAPHY

Gillespie, Daniel T. *Markov Processes: An Introduction for Physical Scientists*, Academic Press 1992, p.47-8

Golomb, Solomon W., *Shift Register Sequences*, Aegean Park Press 1982

Hénon, Michel "A Two-dimensional Mapping with a Strange Attractor", Comm. Math. Physics, 50:69-77, 1976

Heyman, James E., *On the Use of Chaotic Dynamical Systems to Generate Pseudorandom Bitstreams*, Naval Postgraduate School Masters Thesis, March 1993

DISTRIBUTION LIST

Director (2)
Defense Tech Information Center
Cameron Station
Alexandria, VA 22314

Research Office (1)
Code 81
Naval Postgraduate School
Monterey, CA 93943

Library (2)
Code 52
Naval Postgraduate School
Monterey, CA 93943

Professor Richard Franke (1)
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93943

Dr. Jeffery J. Leader, Code MA/Le (10)
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93943

LT James Heyman (1)
PSC 825 Box 58
FPO AE 09627

Dr. Hal Fredricksen, Code MA/Fs (1)
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93943

Dr. Ismor Fischer, Code MA/Fi (1)
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93943

Dr. Todd A. Rovelli (1)
Division of Applied Mathematics
Brown University
Providence, RI 02912

LT Antonio Fontana, Code MA Naval Postgraduate School Monterey, CA 93943	(1)
Dr. Herschel H. Loomis, Jr., Code EC Naval Postgraduate School Monterey, CA 93943	(1)